UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/886,585	06/21/2001	Daniel M. Lavery	2207/11237	6346

7590 06/22/2006

SHARMINI N. GREEN
C/O BLAKLELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER	
RAMPURIA, SATISH	
ART UNIT	**PAPER NUMBER**
2191	

DATE MAILED: 06/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/886,585	LAVERY ET AL.	
	Examiner	Art Unit	
	Satish S. Rampuria	2191	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 March 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Response to Amendment

1. This action is in response to the Amendment received on March 27, 2006.

2. The objection to the specification is withdrawn in view of Applicants' comments in the Remarks and referring to 37 C.F.R. §1.73 does not state "must" or "shall" and the language indicates that it is within the Applicants' discretion to make an election whether to include a summary. Applicants have elected not to include a "Summary of the Invention".

3. Claims amended by the Applicants: 1 and 13.

4. Claims 1-30 are pending.

Response to Arguments

5. Applicant's arguments with respect to claims have been considered but they are not persuasive.

In the remarks, the applicant has argued that:

(i) Applicants submit that there is no suggestion or motivation in either reference for such a combination. The Examiner states that it would have been obvious to combine Torii with Dryfoos to render other elements of the claimed invention unpatentable because "one of ordinary skill in the art would be motivated to select an entry in a trigger table to differentiate between instruction programs and instructions in the operating system's service routine as suggested by Dryfoos". Applicants respectfully submit that this does not suggest a motivation,

merely a result. There is no teaching in either Torii or Dryfoos to actually suggest this combination.

(ii) With respect to claims 22, 24 and 29, Applicants respectfully traverse the Examiner's rejection. Specifically, the section of Dryfoos highlighted by the Examiner makes no mention of creating an entry in a trigger table, the entry associated with the trigger instruction and the auxiliary code, as the Examiner suggests. First and foremost, there is no mention of a trigger table, merely a table. The table in Dryfoos contains "at least one of information identifying at lest some areas of main storage of the computing environment where programs to be debugged may reside, or information identifying at least one program of the computing environment to be excluded from debugging" (Dryfoos, col. 1, lines 58-63). This is not, however, a "trigger table" as claimed. Additionally, Dryfoos merely mentions that the table contains "information", without any mention of creating entries for the table. As such, Applicants maintain that Dryfoos does not teach or suggest the claimed element and that Claim 22 is patentable over the combination of Torri and Dryfoos.

Examiner's response:

(i) In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness

can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988)and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, it is noted that the rejection clearly points out where the combination of Torii and Dryfoos teach the claimed features and why it would have been obvious to combine their teachings. Specifically, the rejection points out that the motivation to "selecting an entry in a trigger table, if the instruction is a trigger instruction, the entry associated with the trigger instruction and entry is referenced by the trigger table" would be to differentiate between instruction in the application programs and instructions in the operating system's service routine. Applicant only makes general allegations of improper hindsight reasoning and does not point out any errors in the rejection. Therefore, the rejection is proper and maintained herein.

Further, for the limitation "executing an auxiliary code", the cited portion of Torii clearly reads on see col. 4, lines 32-33. As indicated previously that as specified in the applicants' specification (paragraph [0012-0015]) that the auxiliary code nothing but executing in parent

child environment. The cited portion by the Examiner is executing the code in parent-child environment. Therefore, the rejection is proper and maintained herein.

(ii) In response to applicant's argument, Dryfoos does teach at least one of information identifying at lest some areas of main storage of the computing environment where programs to be debugged may reside, or information identifying at least one program of the computing environment to be excluded from debugging (col. 1, lines 58-63), as noted by the applicants. So, as broadly interpretation of the limitation the information table could very well be the trigger information table. Therefore, the rejection is proper and maintained herein.

6. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8. Claim 1-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent No. 6,389,446 to Torii, hereinafter called Torii, in view of US Patent No. 6,754,888 to Dryfoos et al., hereinafter called Dryfoos.

Per claim 1:

Torii discloses:

- A method for executing a code (col. 2, lines 56-57 "executes a plurality of threads of instruction streams"), comprising:
- receiving an instruction (col. 6, lines 36-37 "thread manager 5 receives the request");
- determining whether the instruction is a trigger instruction (col. 6, lines 36-42 "determines whether the thread being executed by the one of thread processors... It is determined by checking the content of child thread processor entry number 12 in thread status table 9");
- executing an auxiliary code, the auxiliary code executing separate from the instruction (col. 4, lines 32-33 "a thread generation instruction 2 generates thread

#1 (1b) from thread #0 (1a)"). As understood from the specification (paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Torii does not explicitly disclose selecting an entry in a trigger table, if the instruction is a trigger instruction, the entry associated with the trigger instruction and entry is referenced by the trigger table.

However, Dryfoos, in an analogous computer system discloses selecting an entry in a trigger table, if the instruction is a trigger instruction, the entry associated with the trigger instruction and entry is referenced by the trigger table (col. 1, lines 55-67 to col. 2, lines 1-2 "... defining a table... detecting a debug trigger point during execution of the program, the method further includes referencing the table to ascertain at least one of whether the trigger point is within a program area of the storage of the computing environment, or referencing the table to determine whether the trigger point is included within a program to be excluded from debugging").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of selecting an entry in a trigger table, the entry associated with the trigger instruction and entry is referenced by the trigger table as taught by Dryfoos in the method of executing the code as taught by Torii. The modification would be obvious because of one of ordinary skill in the art would be motivated to select an entry in a trigger table to differentiate between

instruction in the application programs and instructions in the operating system's service routine as suggested by Dryfoos (col. 1, lines 45-52).

Per claim 2:

The rejection of claim 1 is incorporated, and further, Torii disclose:

- spawning a new thread, the new thread executing instructions included in the auxiliary code (col. 4, lines 44-45 "the parent thread of thread #1 (1b) is thread #0 (1a) while the child thread of thread #1 (1b) is thread #2 (1c)"). As understood from the specification (paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment

Per claim 3:

The rejection of claim 2 is incorporated, and further, Torii disclose:

- executing the new thread concurrently with a parent thread, the parent thread including the trigger instruction (col. 4, lines 44-45 "the parent thread of thread #1 (1b) is thread #0 (1a) while the child thread of thread #1 (1b) is thread #2 (1c)").

Per claim 4:

Torii discloses:

- A computer-implemented method for executing a code (col. 2, lines 56-57 "executes a plurality of threads of instruction streams"), comprising:

- receiving a trigger instruction (col. 6, lines 36-37 "thread manager 5 receives the request");
- executing a p-slice code (col. 4, lines 32-33 "a thread generation instruction 2 generates thread #1 (1b) from thread #0 (1a)"). As understood from the specification (See paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Torii does not explicitly disclose selecting an entry in a trigger table, the entry associated with the trigger instruction and entry is referenced by the trigger table.

However, Dryfoos, in an analogous computer system discloses selecting an entry in a trigger table, the entry associated with the trigger instruction and entry is referenced by the trigger table (col. 1, lines 55-67 to col. 2, lines 1-2 "... defining a table... detecting a debug trigger point during execution of the program, the method further includes referencing the table to ascertain at least one of whether the trigger point is within a program area of the storage of the computing environment, or referencing the table to determine whether the trigger point is included within a program to be excluded from debugging").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of selecting an entry in a trigger table, the entry associated with the trigger instruction and entry is referenced by the trigger table as taught by Dryfoos in the method of executing the code as taught by Torii. The modification would be obvious because of one of ordinary skill in the art

would be motivated to select an entry in a trigger table to differentiate between instruction in the application programs and instructions in the operating system's service routine as suggested by Dryfoos (col. 1, lines 45-52).

Per claim 5:

The rejection of claim 4 is incorporated, and further, Torii disclose:

- spawning a new thread, the new thread executing instructions included in the auxiliary code (col. 4, lines 44-45 "the parent thread of thread #1 (1b) is thread #0 (1a) while the child thread of thread #1 (1b) is thread #2 (1c)"). As understood from the specification (paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Per claim 6:

The rejection of claim 5 is incorporated, and further, Torii disclose:

- executing the new thread concurrently with a parent thread, the parent thread including the trigger (col. 4, lines 44-45 "the parent thread of thread #1 (1b) is thread #0 (1a) while the child thread of thread #1 (1b) is thread #2 (1c)").

Per claim 7:

The rejection of claim 6 is incorporated, and further, Torii disclose:

- storing state information from the parent thread before spawning the new thread (col. 11, lines 21-22* "Therefore, thread manager 28 (or 32) stores the child thread start information into thread saving buffer 31 (or 37)").

Per claim 8:

The rejection of claims 7 and 9 are incorporated, respectively, and further, Torii disclose:

- copying the state information for use in the new thread (col. 11, lines 6-11 "Thread information saving line 37 saves a thread execution start address and data essential for starting a child thread's execution...").

Per claim 9:

The rejection of claim 6 is incorporated, and further, Torii disclose:

- storing a register value of the parent thread before spawning the new thread (col. 11, lines 6-11 "Thread information saving line 37 saves a thread execution start address and data essential for starting a child thread's execution...").

Per claim 10:

The rejection of claim 9 is incorporated, and further, Torii disclose:

- copying the register value of the parent thread for use in the new thread (col. 11, lines 6-11 "Thread information saving line 37 saves a thread execution start address and data essential for starting a child thread's execution...").

Per claim 11:

The rejection of claim 4 is incorporated, and further:

- wherein the entry in the trigger table is selected by associative lookup of the trigger instruction. The recited in this claim are similar to those recited in claim 4 and rejected under the same rational set forth in connection with the rejection of claim 4 above.

Per claim 12:

The rejection of claim 4 is incorporated, and further, Torii disclose:

- reading an instruction pointer for the p-slice code from the entry in the trigger table (See FIG. 4, element 9 and related discussion).

Claim 13 is the computer program product claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

Claim 14 is the computer program product claim corresponding to method claim 2 and rejected under the same rational set forth in connection with the rejection of claim 2 above.

Claims 15, 18 and 21 are the system claim corresponding to method claim 1 and rejected under the same rational set forth in connection with the rejection of claim 1 above.

Claims 16 and 19 are the system claim corresponding to method claim 2 and rejected under the same rational set forth in connection with the rejection of claim 2 above.

Claims 17 and 20 are the system claim corresponding to method claim 7 and rejected under the same rational set forth in connection with the rejection of claim 7 above.

Per claims 22 and 24:

Torii discloses:

- A computer-implemented method for compiling (col. 2, lines 56-57 "executes a plurality of threads of instruction streams"), comprising:
- receiving a function body, the function body comprising a trigger instruction (col. 6, lines 36-37 "thread manager 5 receives the request");
- outputting an auxiliary code associated with the function body and the trigger instruction (col. 4, lines 32-33 "a thread generation instruction 2 generates thread #1 (1b) from thread #0 (1a)"). As understood from the specification (See paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Torii does not explicitly disclose creating an entry in a trigger table the entry associated with the trigger instruction and the auxiliary code.

However, Dryfoos, in an analogous computer system discloses creating an entry in a trigger table the entry associated with the trigger instruction and the auxiliary code (col. 1, lines 55-67 to col. 2, lines 1-2 "... defining a table... detecting a debug trigger

point during execution of the program, the method further includes referencing the table to ascertain at least one of whether the trigger point is within a program area of the storage of the computing environment, or referencing the table to determine whether the trigger point is included within a program to be excluded from debugging").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of creating an entry in a trigger table as taught by Dryfoos in the method of executing the code as taught by Torii. The modification would be obvious because of one of ordinary skill in the art would be motivated to select an entry in a trigger table to differentiate between instruction in the application programs and instructions in the operating system's service routine as suggested by Dryfoos (col. 1, lines 45-52).

Per claim 23:

The rejection of claims 22 and 24 is incorporated, respectively, and further, Torii disclose:

- creating a stub block, the stub block comprising a spawn instruction, the spawn instruction configured to spawn a new thread, the new thread configured to execute the auxiliary code (col. 4, lines 44-45 "the parent thread of thread #1 (1b) is thread #0 (1a) while the child thread of thread #1 (1b) is thread #2 (1c)"). It is inherent to create the stub block in order to execute the spawn instructions.

Per claim 24:

Torii discloses:

- A method for compiling (col. 2, lines 56-57 "executes a plurality of threads of instruction streams"), comprising:
- receiving a function body, the function body comprising a trigger instruction (col. 6, lines 36-37 "thread manager 5 receives the request");
- outputting a p-slice code associated with the function body and the trigger instruction (col. 4, lines 32-33 "a thread generation instruction 2 generates thread #1 (1b) from thread #0 (1a)"). As understood from the specification (See paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Torii does not explicitly disclose creating an entry in a trigger table, the entry associated with the trigger instruction and the p-slice code.

However, Dryfoos, in an analogous computer system discloses creating an entry in a trigger table, the entry associated with the trigger instruction and the p-slice code (col. 1, lines 55-67 to col. 2, lines 1-2 "... defining a table... detecting a debug trigger point during execution of the program, the method further includes referencing the table to ascertain at least one of whether the trigger point is within a program area of the storage of the computing environment, or referencing the table to determine whether the trigger point is included within a program to be excluded from debugging").

Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to incorporate the method of creating an entry in a

trigger table as taught by Dryfoos in the method of executing the code as taught by Torii. The modification would be obvious because of one of ordinary skill in the art would be motivated to select an entry in a trigger table to differentiate between instruction in the application programs and instructions in the operating system's service routine as suggested by Dryfoos (col. 1, lines 45-52).

Per claim 25:

The rejection of claim 24 is incorporated, respectively, and further, Torii disclose:

- receiving the p-slice code associated with the function body and the trigger instruction (col. 6, lines 36-37 "thread manager 5 receives the request").

Per claim 26:

The rejection of claim 24 is incorporated, respectively, and further, Torii disclose:

- generating the p-slice code associated with the function body and the trigger instruction (col. 4, lines 32-33 "a thread generation instruction 2 generates thread #1 (1b) from thread #0 (1a)"). As understood from the specification (See paragraph [0012-0015]) the auxiliary code nothing but executing in parent child environment.

Per claim 27:

The rejection of claim 24 is incorporated, respectively, and further:

- creating a stub block, the stub block comprising a spawn instruction, the spawn instruction configured to spawn a new thread, the new thread configured to execute the p-slice code. The recited in this claim are similar to those recited in claim 23 and rejected under the same rational set forth in connection with the rejection of claim 23 above.

Per claim 28:

The rejection of claim 24 is incorporated, respectively, and further, Torii disclose:

- adding store instructions to the stub block, the store instructions configured to store state information of a current (col. 11, lines 6-11 "Thread information saving line 37 saves a thread execution start address and data essential for starting a child thread's execution...") the state information of the current thread including values contained in live-in registers of the new thread (col. 11, lines 24-29 "When thread processor... execution of thread #0, it enters a "free"-state... thread manager 28 (or 32) pulls the child thread start information out of thread saving buffer 31 (or 37), and uploads it into thread processor #0 to start the child thread, or thread #4").

Claims 29 and 30 are the computer program product claim corresponding to method claim 24 and rejected under the same rational set forth in connection with the rejection of claim 24 above.

Conclusion

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to **Satish S. Rampuria** whose telephone number is **(571) 272-3732**. The examiner can normally be reached on **8:30 am to 5:00 pm** Monday to Friday except every other Friday and federal holidays. Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: 571-272-2100**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, **Wei Y. Zhen** can be reached on **(571) 272-3708**. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Satish S. Rampuria
Patent Examiner/Software Engineer
Art Unit 2191

WEI ZHEN
SUPERVISORY PATENT EXAMINER